04-15-04 10715133

सत्यमेव जयते

INTELLECTUAL PROPERTY **INDIA**

GOVERNMENT OF INDIA
MINISTRY OF COMMERCE & INDUSTRY,
PATENT OFFICE, DELHI BRANCH,
W - 5, WEST PATEL NAGAR,
NEW DELHI - 110 008.

I, the undersigned, being an officer duly authorized in accordance with the provision of the Patent Act, 1970 hereby certify that annexed hereto is the true copy of the ***Application, Complete Specification*** *and* ***Drawing Sheets*** *filed in connection with Application for Patent No.****1166/Del/02*** *dated* ***18th November 2002.***

Witness my hand this 12th Day of December 2003.

(S.K. PANGASA)
Assistant Controller of Patents & Designs

FORM 1
THE PATENTS ACT, 1970
(39 of 1970)
APPLICATION FOR GRANT OF A PATENT
(See Sections 5(2), 7, 54 and 135)

1. I/we,

STMicroelectronics Pvt. Ltd., an Indian company, of Plot No. 2 & 3, Sector 16A, Institutional Area, Noida – 201 3001, Uttar Pradesh, India.

2. hereby declare –

   (a) that I am/we are in possession of an invention titled ***"Linearly Scalable Finite Impulse Response Filter."***

   (b) that the ~~provisional~~/ complete specification relating to this invention is filed with this application

   (c) that there is no lawful ground of objection to the grant of a patent to me/us.

3. further declare that the inventor(s) for the said inventions is/are

   (i) SAHA Kaushik, an Indian citizen, of A2, Staff Flats, I.P. College, Shamnath Marg, Delhi – 110 054, India.

   (ii) MAITI Srijib Narayan, an Indian citizen, of R-5/1, Duk Bungalow Road, Saratpalli, Midnapore – 721101, W.B. India.

4. I/we claim the priority from the application(s) filed in convection countries, particulars of which are as follows: **NA**

5. I/we state that the said invention is an improvement in or modification of the invention the particulars of which are as follows and of which I/we are the applicant/patentee: **NIL**

6. I/we state that the application is divided out of my/our application, the particulars of which are given below and pray that this application be deemed to have been filed on under section 16 of the Act. **NIL**.

7. That I am/we are the assignee or legal representative of the true and first inventors.

8. That my/our address for service in India is as follows:

   ANAND & ANAND, Advocates
   B-41, Nizamuddin East
   New Delhi – 110 013

   Tel Nos.: (11) 4355078, 4355076, 4350360
   Fax Nos.: (11) 4354243, 4352060

9- I/We the true and first inventors of this invention or the applicant(s) in the convention country declare that the applicant(s) herein is/are my/our assignee or legal representative.

a) kaushik saha an Indian National of *A2, Staff Flats, I.P. College, Shamnath Marg, Delhi-110054, India.*

Signature

Kaushik Saha

Dated this 18th day of November 2002

b) *Srijib Narayan Maiti* an Indian National of *R-5/1, Duk Bungalow Road, Saratpalli, Midnapore-721101, W.B., India.*

Signature

Srijib Narayan Maiti

Dated this 18th day of November 2002

10- that to the best of my/our knowledge, information and belief the fact and matters stated herein are correct and that there is no lawful ground of objection to grant of patent to me/us on this application.

11- Following are the attachment with the application

(a) Complete specification (3 copies)
(b) Abstract
(c) Formal drawings
(d) Power of Attorney
(e) Form 1 (in triplicate)
(f) Form 3 (in duplicate)
(g) Fee Rs. 5000/- In cash/cheque/bank draft bearing no. , date On Bank.

I/We request that a patent may be granted to me/us for the said invention.

Dated this 18th day of November 2002

Signature
STMicroelectronics Pvt. Limited

To

The Controller of Patents

The Patent Office, Delhi

02-IND-205

Form 2

1100-DEL-02

THE PATENTS ACT, 1970

18 NOV 2002

COMPLETE SPECIFICATION

[See Section 10]

'LINEARLY SCALABLE FINITE IMPULSE RESPONSE FILTER'

ORIGINAL

STMicroelectronics Pvt. Ltd., *Plot No. 2 & 3, Sector 16A, Institutional Area, Noida – 201 301, Uttar Pradesh, India, an Indian Company*

The following specification particularly describes and ascertains the nature of this invention and the manner in which it is to be performed.

LINEARLY SCALABLE FINITE IMPULSE RESPONSE FILTER

Field of the Invention:

The invention relates to Finite Impulse Response (FIR) Digital Filters. More specifically the invention relates to an efficient implementation for Finite Impulse Response filters in multi-processor architecture.

Background of the invention:

Finite Impulse Response (FIR) filters are important and widely used form of digital filter. FIR filters are used in numerous real time applications including as telecommunication, particularly for the generation of constant phase delay characteristics. In addition, signal processing applications requiring interpolation and decimation function incorporate FIR filtration as an integral part of the process.

The output sample of a FIR filter is a convolution summation of input samples and the impulse response of the filter. The output, y(n).of a causal FIR filter can be written as:

$$y(n) = \sum_{k=0}^{H-1} h(k) * x(n-k) \qquad \text{-------} (1)$$

Where:

H is the total number of filter coefficients and **n** = 0, 1, 2, 3... for different values of **n**, the output sample of the filter can be obtained.

h(k) is the impulse response of the filter. Filter coefficients are determined for various values of **k**. The value of **k** can never be negative for a causal system.

x(m) is the input sample, **m** = **n** - **k** as shown in equation (1). The value of **m** can never be negative for a casual system.

As stated by the above equation, the coefficients are multiplied with the appropriate input samples and summed to obtain the output sample.

The coefficients are multiplied with the appropriate input samples and then accumulated for obtaining a particular output sample.

For N number of input samples and H number of coefficients, the required number of multiplications for a given output sample is H. The saturation occurs at the H^{th} output sample as shown in **figure 1.**

The H number of multiplications are necessarily required if all the H coefficients are unique.

Compared to other filters, FIR filters offer the following advantages:
FIR filters are simple to implement and design. Linear-phase filters that delay the input signal, without causing phase distortion, can be easily realized using FIR filters. FIR filters do not employ any feedback and hence present fewer difficulties in practical implementation. The absence of feedback also simplifies design by making it possible to use finite precision arithmetic. Multi-rate applications such as "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), can be realized best by FIR filters.

FIR filters can be easily implemented using fractional arithmetic unlike other type of filters in which it is difficult to do so. It is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0 as the overall gain of the FIR filter can be adjusted at its output. All the above advantages make FIR filters preferable for fixed-point Digital Signal Processors (DSPs).

The conventional approach of FIR filter implementation utilizes a delay line ($\mathbf{m = n - k}$ eq. 1), resulting in increased memory requirements and slower computation.

US patent **5,732,004** describes an algorithm for FIR filtering. It proposes a method of decimating and/or interpolating a multi-bit input signal in which n/2 additions are performed, where 'n' is the number of bits in each filter coefficient. Scaling and multiplication of data with coefficients is performed using standard DSP architecture using coefficient values and associated scaling factors stored in memory. The coefficients are stored in coded form, and are then decoded prior to multiplication by the data values. A delay line is essential for the implementation of this method.

US patents **6,260,053** and **5,297,069** describe methods and implementations that utilize delay lines for FIR filter implementation. Moreover, none of these methods provides linear scalability.

The Objects and Summary of the invention:

The object of the invention is to provide an efficient implementation for FIR filter for multi-processor architectures.

Another object of this invention is to provide a linearly scalable FIR filter.

Yet another object of the invention is to obviate the need of a delay line for implementing an FIR filter thereby reducing the memory requirement for computation and resulting in a cost effective solution.

It is a further object of the invention to speed up the computation of FIR filters.

To achieve said objective this invention provides an improved Finite Impulse Response (FIR) filter providing linear scalability and implementation without the need for delay lines, comprising, a multiprocessor architecture including a plurality of ALUs (Arithmetic and Logic Unit), Multipliers units, Data cache, and Load/Store units sharing a common Instruction cache, and multi-port memory, and an assigning means for assigning to each available processing unit the computation of specified unique partial

product terms and the accumulation of each computed partial product on specified output sample values.

Wherein the assigning means is a pre-process that is used to design the implementation of the FIR filter based on the filter specifications.

Further the invention provides a method for implementing an improved Finite Impulse Response (FIR) filter providing linear scalability using a multiprocessing architecture platform without the need for delay lines, comprising the steps of:

- assigning to each available processing unit the computation of specified unique partial product terms and
- the accumulation of each computed partial product on specified output sample values.

Brief Description of drawings and set of equations:

The invention will now be described with reference to the accompanying drawings:

Figure 1 shows a generalized set of equations for the outputs of an FIR filter with N input/output samples, and H filter coefficients.

Figure 2 shows a set of equations for the outputs of a FIR filter having 11 input/output samples, and 6 filter coefficients.

Figure 3 shows a set of equations for the outputs of an FIR filter having 13 input/output samples, and 7 filter coefficients.

Figure 4 shows a schematic diagram of a Multi-processor architecture.

Figure 5 shows a schematic diagram of the internal architecture of one processor of a multi-processor architecture.

Detailed Description of the invention

Figure1 shows a generalized set of equations for the outputs of an FIR filter with N input/output samples, and H filter coefficients. These equations define the outputs from an FIR filter.

Figure 2 refers to an example of the output samples of a symmetric FIR filter having 11 input/output samples, and 6 filter coefficients. If the total number of coefficients 'H' is an even number as shown in this example, then it is possible to obtain all the output samples by computing a subset of partial products that involve 'H/2' coefficients. Each of these partial products occurs in 2 symmetrically spaced samples, and can therefore be computed once and reused thereby eliminating the requirement of a delay line and reducing computational time and memory space requirements, as compared to the conventional approach of FIR filter implementation.

In the example of **Figure 2** output sample y(5) requires the computation of only 3 partial products i.e. x(5)h(0), x(4)h(1) and x(3)h(2) as the other three partial product terms x(0)h(0), x(1)h(1) and x(2)h(2) have already been computed in previous samples y(0), y(2) and y(4) respectively. Similarly, x(5)h(0), x(4)h(1) and x(3)h(2) contribute to terms y(10), y(8) and y(6) respectively and so on. It can be observed from **figure 2** that 'P' number of consecutive output calculations can be distributed over 'P' processors simultaneously, provided 'P' consecutive inputs are present. As shown for the case of two processors ($P = 2$), y(5) and y(6) can be calculated simultaneously.

Figure 3 shows an example of the output samples of a symmetrical FIR filter having 13 input/output samples, and 7 filter coefficients. If the total number of coefficients 'H' is an odd number as shown in this example, then it is possible to obtain all the output sample by calculating a maximum ((H-1)/2+1) number of partial products after the saturation. When the filter coefficient are odd i.e. H is odd, the basic algorithm remains the same as

for the even number of coefficients case with the exception for the $((H-1)/2 + 1)^{th}$ column around which the **figure 3** is centro-symmetric. The total number of the required multiplications for each output sample is $((H-1)/2 + 1)$ after the saturation. As shown, the product term associated with h(3) can not be reused to calculate the partial summation for any other output sample.

The basic algorithm for symmetric FIR filters computation is as follows:

- Initialize loop index by zero
- Load input sample and coefficients
- Multiply input sample and coefficient
- Update current output
- Update partial output, if required (see **Note** below)
- Increment loop index by Number of processor
- If loop termination is condition satisfied then stop, else go to second step.

Note: For Centro-symmetric case one product term is never re-used.

For the case of a **2 parallel-processors** architecture the basic algorithm is modified as follows:

- Initialize loop-index by zero
- Load input sample for processor #1
- Load coefficient for processor #1
- Multiply input sample and coefficient in processor #1
- Update current output in processor #1
- Update partial output, if required, in processor #1
- Load input sample for processor #2
- Load coefficient for processor #2
- Multiply input sample and coefficient in processor #2
- Update current output in processor #2
- Update partial output, if required, in processor #2
- Increment loop-index by 2

- If loop-index equals to (N*H) then stop, or go to second step

The algorithm can be generalized for any number of processors in parallel.
Further the algorithm is also applicable for the asymmetric FIR Filters in following steps.

- Initialize loop index by zero
- Load input sample and coefficients
- Multiply input sample and coefficient
- Update current output
- Increment loop index by Number of processor
- If loop termination is condition satisfied then stop, else go to second step.

Each processor in the parallel processing architecture is provided with an independent ALU (Arithmetic and Logic Unit), Multiplier unit, Data cache, and Load/Store unit. Al the processors share a common Instruction cache, and multi-port memory. This architecture is typical of VLIW (Very Large Instruction Word) processors.

Figure 4 shows the typical architecture of a Very Long Instruction Word (VLIW) processor core. It is a Multi-processor architecture in which each processors is composed of a mix of Register Banks, Constant Generators (immediate operands) and Functional Units. Different processors may have different unit/register mixes, but a single Program Counter and a unified I-cache controls them all, so that all processors run in lockstep (as expected in a VLIW). Similarly, the execution pipeline drives all processors. Inter-processor communication, achieved by explicit register-to-register move, is compiler-controlled and invisible to the programmer. At the multi-processor level, the typical architecture specification for such a device is as follows:

- An instruction delivery mechanism is provided to get instructions from the cache to the processors' data-path.
- An inter-processor communication mechanism is provided to transfer data among processors.
- The data-cache is organized to establish a guarantee of main memory coherency in the presence of multiple memory accesses.

The Algorithm to compute output samples resides in the instruction fetch cache and expansion unit.

Figure 5 shows the schematic diagram of the internal architecture of the processor. Generally a processor is a 4-issue (maximum 4 instructions can be issued simultaneously) VLIW core comprising the following:

Four 32-bit integer ALUs, two 16x32 multipliers, one Load/Store Unit and one Branch Unit.

Sixty four 32-bit General-purpose registers and 8 1-bit branch registers (used to store branch condition, predicates and carries). Instructions allow two long immediate data operands per cycle.

The Instruction Set Architecture is a very simple integer RISC instruction set with minimal."predication" support through select instructions.

The memory-addressing repertoire includes base + offset addressing, allows speculative execution (dis-missible loads, handled by the protection unit) and software pre-fetching.

The instant invention utilizes this architecture to provide faster computation as it is evident from the fact that, if any FIR filter takes T units of time to compute N outputs in one processor, then for the increased number of processors (say P), the time required to compute N outputs will be T/P units of time, which provides 'Linear scalability'. Further, since all the processors load input samples and coefficients from the shared memory where both the data are available, there is no requirement for any delay line. The instant invention does not limit its scope to non-unique filter coefficients, but both the non-utilization of delay line as well as linear scalability can be realized for all unique coefficients.

We claim:

1. An improved symmetric Finite Impulse Response (FIR) filter providing linear scalability and implementation without the need for delay lines, comprising:
   - a multiprocessor architecture including a plurality of ALUs (Arithmetic and Logic Unit), Multipliers units, Data cache, and Load/Store units sharing a common Instruction cache, and multi-port memory, and
   - an assigning means for assigning to each available processing unit the computation of specified unique partial product terms and the accumulation of each computed partial product on specified output sample values.

2. An improved FIR filter, as claimed in claim 1 wherein the assigning means is a preprocess that is used to design the implementation of the FIR filter based on the filter specifications.

3. A method for implementing an improved symmetric Finite Impulse Response (FIR) filter providing linear scalability using a multiprocessing architecture platform without the need for delay lines, comprising the steps of:
   - assigning to each available processing unit the computation of specified unique partial product terms and the accumulation of each computed partial product on specified output sample values.

4. An improved symmetric Finite Impulse Response (FIR) filter providing linear scalability and implementation without the need for delay lines substantially as herein described with reference to and as illustrated in the accompanying drawings.

5. A method for implementing an improved symmetric Finite Impulse Response (FIR) filter providing linear scalability using a multiprocessing architecture platform without the need for delay lines substantially as herein described with reference to and as illustrated in the accompanying drawings

Dated this 18th day of November 2002

Shanti Kumar

of **ANAND & ANAND, Advocates**
Agents for the Applicants

ABSTRACT

The present invention provides an improved Finite Impulse Response (FIR) filter providing linear scalability and implementation without the need for delay lines, comprising, a multiprocessor architecture including a plurality of ALUs (Arithmetic and Logic Unit), Multipliers units, Data cache, and Load/Store units sharing a common Instruction cache, and multi-port memory, and an assigning means for assigning to each available processing unit the computation of specified unique partial product terms and the accumulation of each computed partial product on specified output sample values.

Further the invention also provides a method for implementing an improved Finite Impulse Response (FIR) filter providing linear scalability using a multiprocessing architecture platform without the need for delay lines.

$$y(0) = x(0)h(0)$$
$$y(1) = x(1)h(0) + x(0)h(1)$$
$$y(2) = x(2)h(0) + x(1)h(1) + x(0)h(2)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$y(H-1) = x(H-1)h(0) + x(H-2)h(1) + x(H-3)h(2) + \ldots\ldots\ldots +x(0)h(H-1)$$

----saturation from this point onwards

$$y(H) = x(H)h(0) + x(H-1)h(1) + x(H-2)h(2) + \ldots\ldots\ldots +x(1)h(H-1)$$
$$y(H+1) = x(H+1)h(0) + x(H)h(1) + x(H-1)h(2) + \ldots\ldots\ldots +x(2)h(H-1)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$y(N-1) = x(N-1)h(0) + x(N-2)h(1) + x(N-3)h(2) + \ldots\ldots +x(N-H+1)h(H-1)$$

Figure 1

Shanti Kumar
Of Anand And Anand Advocates
Attorney for the Applicant

$$
\begin{aligned}
y(0) &= x(0)h(0)\\
y(1) &= x(1)h(0) + x(0)h(1)\\
y(2) &= x(2)h(0) + x(1)h(1) + x(0)h(2)\\
y(3) &= x(3)h(0) + x(2)h(1) + x(1)h(2) + x(0)h(2)\\
y(4) &= x(4)h(0) + x(3)h(1) + x(2)h(2) + x(1)h(2) + x(0)h(1)\\
y(5) &= x(5)h(0) + x(4)h(1) + x(3)h(2) + x(2)h(2) + x(1)h(1) + x(0)h(0)\\
y(6) &= x(6)h(0) + x(5)h(1) + x(4)h(2) + x(3)h(2) + x(2)h(1) + x(1)h(0)\\
y(7) &= x(7)h(0) + x(6)h(1) + x(5)h(2) + x(4)h(2) + x(3)h(1) + x(2)h(0)\\
y(8) &= x(8)h(0) + x(7)h(1) + x(6)h(2) + x(5)h(2) + x(4)h(1) + x(3)h(0)\\
y(9) &= x(9)h(0) + x(8)h(1) + x(7)h(2) + x(6)h(2) + x(5)h(1) + x(4)h(0)\\
y(10) &= x(10)h(0) + x(9)h(1) + x(8)h(2) + x(7)h(2) + x(6)h(1) + x(5)h(0)
\end{aligned}
$$

Figure 2

Shanti Kumar

Of Anand And Anand Advocates
Attorney for the Applicant

$$
\begin{aligned}
y(0) &= x(0)h(0)\\
y(1) &= x(1)h(0) + x(0)h(1)\\
y(2) &= x(2)h(0) + x(1)h(1) + x(0)h(2)\\
y(3) &= x(3)h(0) + x(2)h(1) + x(1)h(2) + x(0)h(3)\\
y(4) &= x(4)h(0) + x(3)h(1) + x(2)h(2) + x(1)h(3) + x(0)h(2)\\
y(5) &= x(5)h(0) + x(4)h(1) + x\text{[illegible]})h(2) + x(2)h(3) + x(1)h(2) + x(0)h(1)\\
y(6) &= x(6)h(0) + \text{[illegible]})(1) \text{[illegible]}h(2) + x(3)h(3) + x(2)h(2) + x(1)h(1) - x(0)h(0)\\
y(7) &= x(7)h(0) + x(6)\text{[illegible]} + x(5)h(2) + x(4)h(3) + x(3)h(2) + x(2)h(1) - x(1)h(0)\\
y(8) &= x(8)h(0) + x(7)h(1) + x(6)h(2) + x(5)h(3) + x(4)h(2) + x(3)h(1) - x(2)h(0)\\
y(9) &= x(9)h(0) + x(8)h(1) + x(7)h(2) + x(6)h(3) + x(5)h(2) + x(4)h(1) - x(3)h(0)\\
y(10) &= x(10)h(0) + x(9)h(1) + x(8)h(2) + x(7)h(3) + x(6)h(2) + x(5)h(1) - x(4)h(0)\\
y(11) &= x(11)h(0) + x(10)h(1) + x(9)h(2) + x(8)h(3) + x(7)h(2) + x(6)h(1) + x(5)h(0)\\
y(12) &= x(12)h(0) + x(11)h(1) + x(10)h(2) + x(9)h(3) + x(8)h(2) + x(7)h(1) - x(6)h(0)
\end{aligned}
$$

No reusability

2 1

Figure 3

BEST AVAILABLE COPY

02-IND-205

Shanti Kumar
Of Anand And Anand Advocates
Attorney for the Applicant

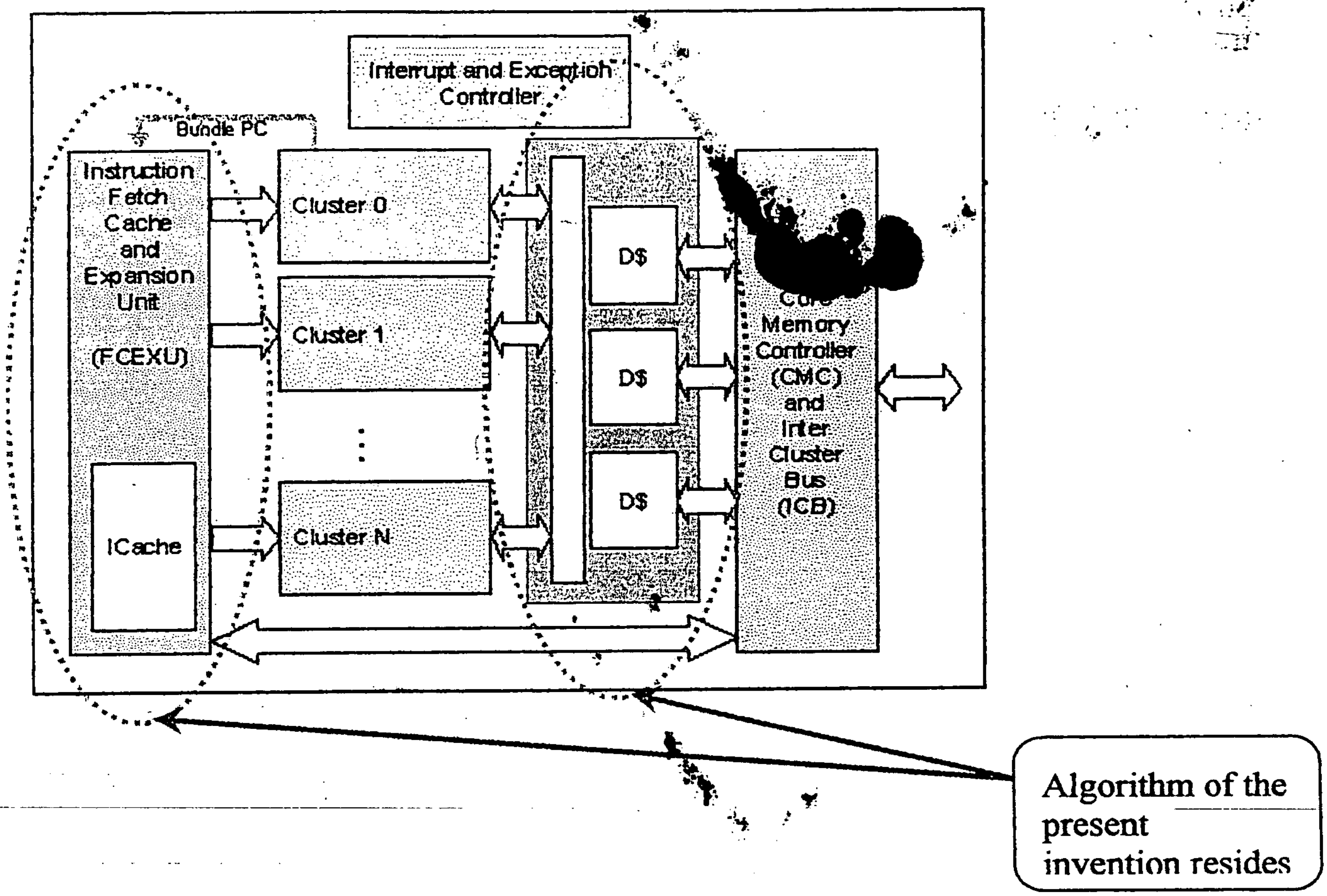

Figure 4

BEST AVAILABLE COPY

Shanti Kumar
Of Anand And Anand Advocates
Attorney for the Applicant

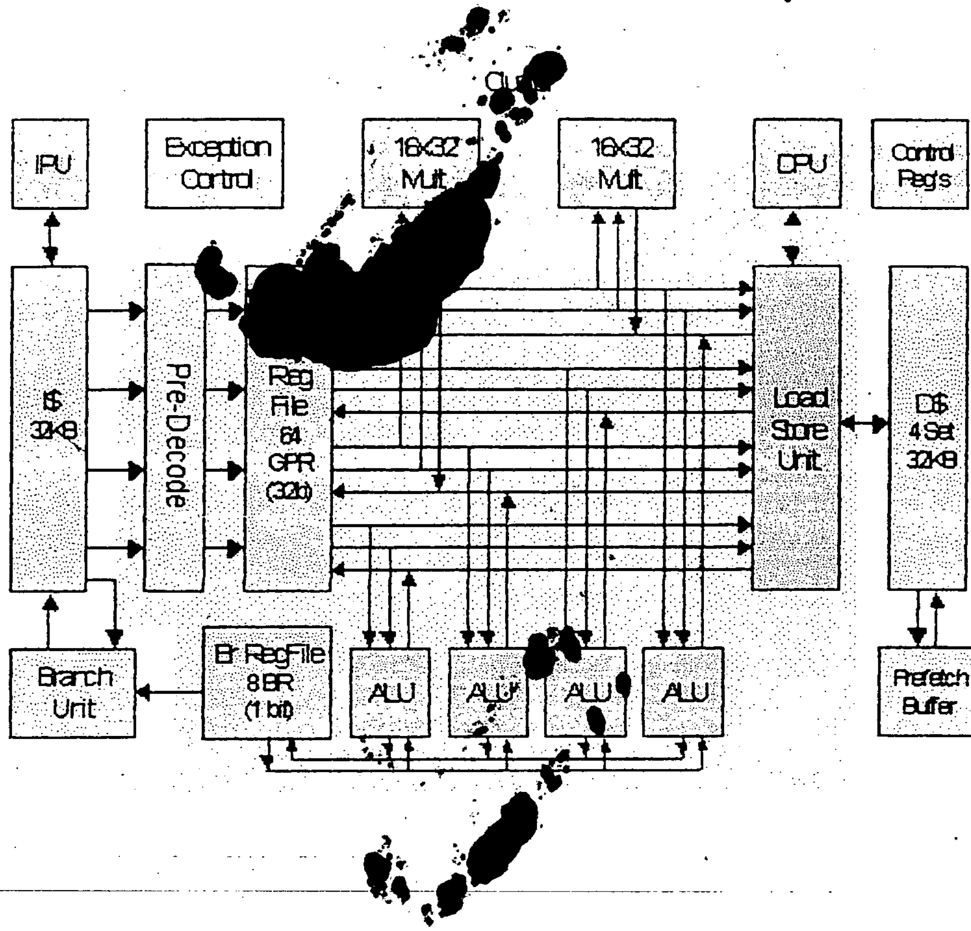

Figure 5

BEST AVAILABLE COPY

Shanti Kumar
Of Anand And Anand Advocates
Attorney for the Applicant